

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019REMARKSRECEIVED
CENTRAL FAX CENTERI. INTRODUCTION

JUL 23 2007

Claims 1-60 are pending in the present application. Claims 1, 9, 16, 23, 36-39 and 55 have been amended. In view of the above amendments and the following remarks, it is respectfully submitted that all of the pending claims are allowable.

II. CLAIM REJECTIONS – 35 U.S.C. § 101

In the Decision on Appeal, the Board *sua sponte* rejected claims 1-15, 40-41 and 43-56 under 35 U.S.C. § 101 as directed to non-statutory subject matter. (See 5/23/07 Decision on Appeal, pp. 4, 12-31.)

In view of the amendments to claim 1, from which claims 2-8, 40-41 and 43-54 depend, to claim 9, from which claims 10-15 depend, and to claim 55, from which claim 56 depends, it is respectfully submitted that this rejection should be withdrawn.

III. CLAIM REJECTIONS – 35 U.S.C. § 102(b)

In the Final Office Action, the Examiner rejected claims 1-41 and 43-60 under 35 U.S.C. § 102(b) as anticipated by John Levine, “Linkers and Loaders, chapter 6,” June 1999 (hereinafter “Levine”). (See 8/25/05 Office Action, pp. 3-10.) This rejection was reiterated in the Advisory Action and affirmed by the Board of Patent Appeals and Interferences in the Decision on Appeal (See 2/6/06 Office Action, pp. 1-2; 5/23/07 Decision on Appeal, pp. 4-11.)

Levine generally describes a technique used by an archive format (i.e., the “ar” command) for creating libraries. A library is handled by a program linker to resolve symbol references within a program. (See Levine, “Library formats,” p. 1.) The libraries may contain selected object files, or routines, that resolve undefined symbols, and these object files may be

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019

used by linkers and loaders in order to automate symbol resolution. (See id.) In addition, these collections of object files within the libraries may be created using various formats, wherein the simplest format for a library is a sequence of object modules. (See id.) The format of the linker libraries implemented in UNIX and Windows is an “archive” format, which may be used for collections of any types of files. (See id.)

For the creation of libraries, the archive formats may use a variety of techniques depending on the support provided by a given operating system. (See id. at “Creating libraries,” p. 5.) In order to deal with the issue of ordering the object files within a created archive library, UNIX systems contains two programs, “lorder” and “tsort,” to help in the creation process. (See id.) Lorder is used to produce a dependency list of the object files that reference specific symbols in other object files. (See id.) In other words, Lorder may provide a list of functions that are required by other functions. The creation of the list is accomplished by extracting the symbols using a symbol listing utility, text-processing the symbols, and using standard sort and join utilities to create an output. (See id.) Tsort performs a topological sort on the output in order to produce a sorted list of object files. (See id.) Each symbol may be defined after all the references to it, thereby allowing all undefined references to be resolved over a single sequential pass. (See id.) Therefore, an archive library may be created with the lorder and tsort programs where the output of lorder is used to control the archive library, “ar”. (See id.) Accordingly, lorder and tsort may be used to reorder the dependencies of symbols within an archive library in order to find all external references.

As amended, claim 1 recites “[a] method, comprising: receiving a software module, the software module including references to locations within the software module, at least some of the references being backward references; *reordering components of the software module into a predetermined order to remove at least some of the backward references*; and displaying an output of the software module to a user, *wherein the components include at least one of a header, a section, and a table.*”

Applicant respectfully submits that Levine does not disclose “reordering components of the software module into a predetermined order,” as recited in claim 1. Further, Applicant

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019

respectfully submits that Levine does not disclose "wherein the components include at least one of a header, a section, and a table," as recited in claim 1. Accordingly, the rejection of claim 1 should be withdrawn. Because claims 2-8 and 40-41 and 43-54 depend from, and, therefore, include all of the limitations of claim 1, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

As amended, claim 9 recites "[a] system, comprising: a reorder module configured to receive a software module including references to locations within the software module, at least some of the references being backward references, *the reorder module configured to reorder components of the software module into a predetermined order* and remove at least some of the backward references, *the components including at least one of at least one of a header, a section, and a table*; and a display to display at least an output of the software module to a user." It is respectfully submitted that Levine does not disclose every element of claim 9, as amended, for the reasons discussed above with reference to claim 1. Accordingly, the rejection of claim 9 should be withdrawn. Because claims 10-15 depend from, and, therefore, include all of the limitations of claim 9, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 38, as amended, recites "[a]n article of manufacture comprising a computer-readable medium having stored thereon instructions adapted to be executed by a processor, the instructions which, when executed, define a series of steps to be used to reorder a software module, said steps comprising: receiving a software module, the software module including references to locations within the software module, at least some of the references being backward references; and reordering the components of the software module into a predetermined order to remove at least some of the backward references, wherein the components include at least one of a header, a section, and a table." It is respectfully submitted that Levine does not disclose every element of claim 38, as amended, for the reasons discussed above with reference to claim 1. Accordingly, the rejection of claim 38 should be withdrawn.

Claim 55, as amended, recites "A method, comprising: receiving a software module, the software module including components arranged in a first order, a first one of the components

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019

including a reference to a location in a second one of the components, the second one of the components preceding the first one of the components in the first order; arranging the components into a predetermined second order so that the second one of the components is subsequent to the first one of the components in the second order; and displaying an output of the software module to a user, wherein the components include at least one of a header, a section, and a table.” It is respectfully submitted that Levine does not disclose every element of claim 55, as amended, for the reasons discussed above with reference to claim 1. Accordingly, the rejection of claim 55 should be withdrawn. Because claims 56-60 depend from, and, therefore, include all of the limitations of claim 55, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 16, as amended, recites “[a] method, comprising: receiving a software module sequentially, the software module having at least one symbol reference; *locating a section header table of the software module*; linking the software module onto a target memory space; and *resolving the at least one symbol reference, using the section header table, without storing the entire software module in local memory while the symbol reference is resolved.*”

Applicant respectfully submits that Levine does not disclose “locating a section header table of the software module,” as recited in claim 16. Further, Applicant respectfully submits that Levine does not disclose “resolving the at least one symbol reference, using the section header table, without storing the entire software module in local memory while the symbol reference is resolved,” as recited in claim 16. Accordingly, the rejection of claim 16 should be withdrawn. Because claims 17-22 depend from, and, therefore, include all of the limitations of claim 16, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 23, as amended, recites “A system, comprising: a linker configured to sequentially receive a software module having at least one symbol reference, the linker configured to locate a section header table of the software module, the linker configured to resolve the symbol reference using at least the section header table, the linker configured to store less than the entire software module in local memory during the resolution of the at least one symbol reference.” It

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019

is respectfully submitted that Levine does not disclose every element of claim 23, as amended, for the reasons discussed above with reference to claim 16. Accordingly, the rejection of claim 23 should be withdrawn. Because claims 24-35 depend from, and, therefore, include all of the limitations of claim 23, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 36, as amended, recites “[a] computer readable storage medium including a set of instructions representing a software module that is executable by a processor, the set of instructions operable to: receive a software module sequentially, the software module having at least one symbol reference; locate a section header table of the software module; link the software module onto a target memory space; and resolve the at least one symbol reference, using at least the section header table, without storing the entire software module in local memory while the symbol reference is resolved.” It is respectfully submitted that Levine does not disclose every element of claim 36, as amended, for the reasons discussed above with reference to claim 16. Accordingly, the rejection of claim 36 should be withdrawn. Because claims 37 depends from, and, therefore, includes all of the limitations of claim 36, it is respectfully submitted that this claim is also allowable for at least the reasons stated above.

Claim 39, as amended, recites “[a]n article of manufacture comprising a computer-readable medium having stored thereon instructions adapted to be executed by a processor, the instructions which, when executed, define a series of steps to be used to control the linking of a software module, said steps comprising: receiving a software module sequentially, the software module having at least one symbol reference; locating a section header table of the software module; linking the software module onto a target memory space; and resolving the at least one symbol reference, using at least the section header table, without storing the entire software module in local memory at one time.” It is respectfully submitted that Levine does not disclose every element of claim 39, as amended, for the reasons discussed above with reference to claim 16. Accordingly, the rejection of claim 39 should be withdrawn.

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019

RECEIVED
CENTRAL FAX CENTER

JUL 23 2007

IV. CLAIM REJECTIONS – 35 U.S.C. § 103(a)

In the Final Office Action, the Examiner rejected claim 42 as unpatentable over Levine in view of U.S. Patent No. 6,185,733 to Breslau et al. (hereinafter “Breslau”). (See 8/25/05 Office Action, pp. 10-11.) This rejection was reiterated in the Advisory Action and affirmed by the Board of Patent Appeals and Interferences in the Decision on Appeal (See 2/6/06 Office Action, pp. 1-2; 5/23/07 Decision on Appeal, pp. 11-12.)

Breslau describes a library search process within a computer linkage program. (See Breslau, col. 1, ll. 15-17.) The linkage editor performs the editing on the linkage editor statements and the object modules to be linked. (See *id.*, col. 3, ll. 50-55.) The object modules to be linked may reside in a remote object module or search library or remote system. (See *id.*, col. 4, ll. 11-20.)

However, Applicant respectfully submits that Breslau does not cure the deficiencies of Levine discussed above with reference to claim 1. Therefore, Levine and Breslau, alone or in combination, neither disclose nor suggest every element of claim 1. Accordingly, Applicant respectfully submits that the rejection of claim 42, which depends from claim 1, should be withdrawn.

Attorney Docket No.: 40101/06901
Ref. No.: 2000.019

CONCLUSION

RECEIVED
CENTRAL FAX CENTER

JUL 23 2007

It is therefore respectfully submitted that all of the presently pending claims are allowable. All issues raised by the Examiner having been addressed, an early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

By: 
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, New York 10038
Tel.: (212) 619-6000
Fax: (212) 619-0276

Dated: July 23, 2007